



**PUC Minas**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**Algoritmos e Estrutura de Dados**

**RELATÓRIO TÉCNICO SOBRE APLICAÇÃO DE GERENCIAMENTO DE DADOS**

Bruno Gustavo Rezende

Cleber Gomes Junior

Eric Carvalho Pedro

Gabriel Eduardo Ramos Vignoli

Victor Lucas de Ângela Martins

Betim

2024

## 1. Introdução

Este relatório descreve o desenvolvimento de uma aplicação para gerenciar dados de pessoas, com base em arquivos CSV. O objetivo do projeto foi criar uma aplicação com interface gráfica amigável ao usuário, capaz de carregar e manipular diferentes tamanhos de arquivos CSV, além de fornecer funcionalidades de CRUD (Create, Read, Update, Delete).

## 2. Objetivos

Os principais objetivos do projeto foram:

- Desenvolver uma aplicação com interface gráfica amigável.
- Implementar funcionalidades de CRUD para gerenciar dados de pessoas.
- Permitir a pesquisa e ordenação de pessoas com base em critérios definidos.

## 3. Metodologia

### 3.1 Escolha das Tecnologias

Para atender aos requisitos do projeto, optamos por utilizar a linguagem de programação C#, juntamente com o banco de dados MySQL. Essa escolha foi feita devido à familiaridade da equipe com essas tecnologias e à sua adequação para o desenvolvimento.

### 3.2 Desenvolvimento da Aplicação

O desenvolvimento da aplicação foi dividido em várias etapas:

1. **Análise dos Requisitos:** Compreensão dos requisitos do projeto e definição das funcionalidades a serem implementadas.
2. **Implementação da Interface Gráfica:** Desenvolvimento da interface gráfica da aplicação, seguindo as diretrizes de usabilidade e design sugeridas.
3. **Manipulação de Dados:** Implementação das funcionalidades de carga de arquivos CSV, CRUD de pessoas, pesquisa e ordenação.
4. **Utilização de Estruturas de Dados:** Escolha e implementação das estruturas de dados adequadas para armazenar e manipular os dados de pessoas de forma eficiente.

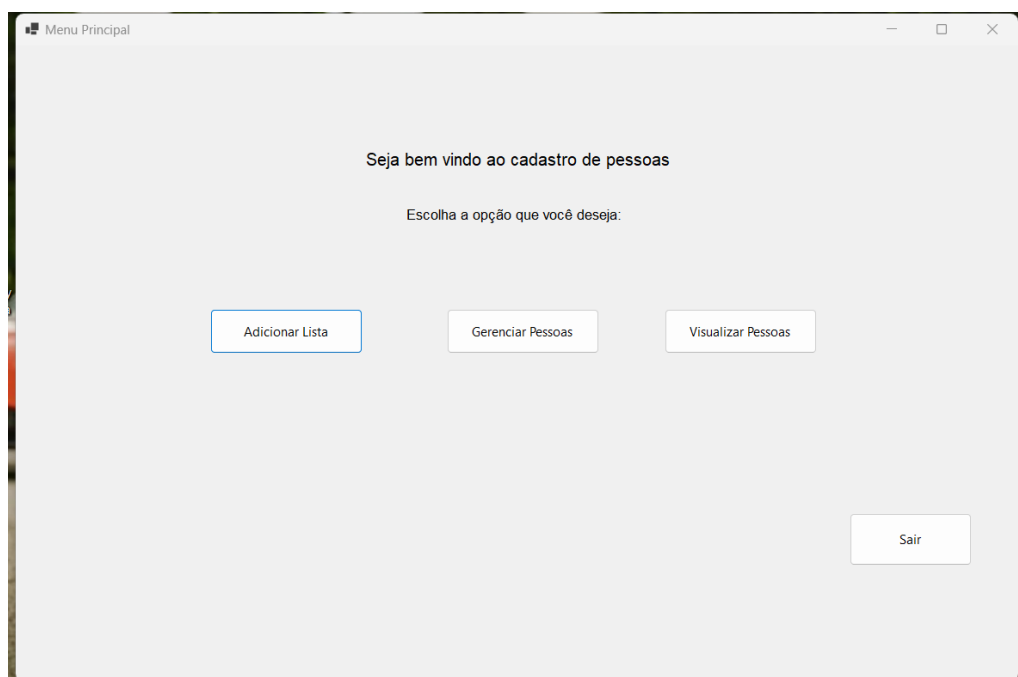
## 4. Funcionalidades Implementadas

A aplicação desenvolvida possui as seguintes funcionalidades:

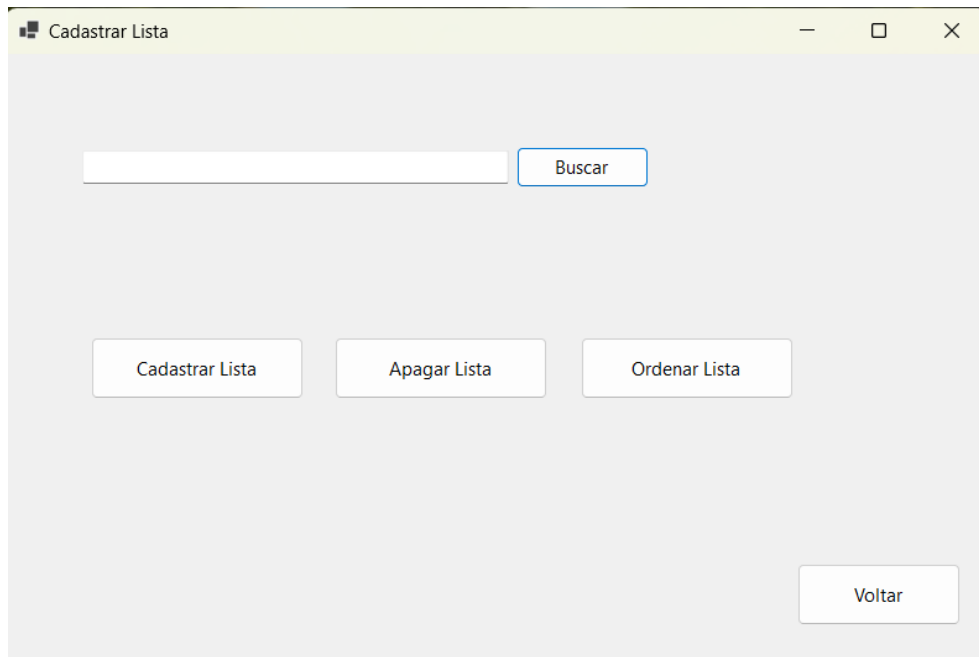
### 4.1 Interface Gráfica

A interface gráfica da aplicação apresenta um menu principal com as seguintes opções:

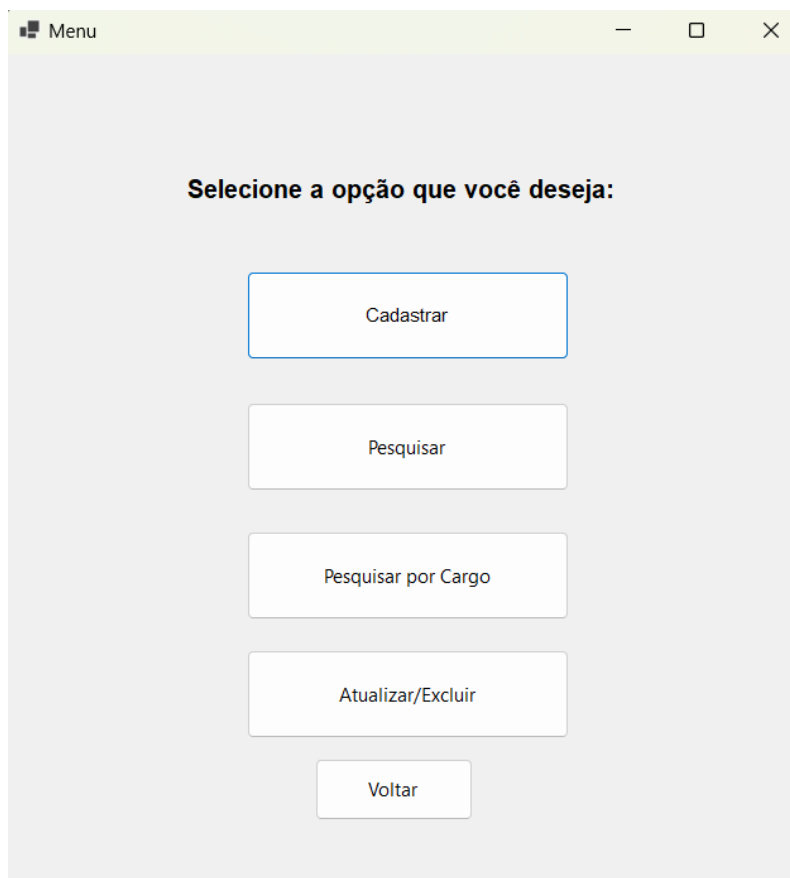
- **Menu principal:** Permite adicionar uma lista de pessoas, gerenciar pessoas, visualizar pessoas, além de sair programa.



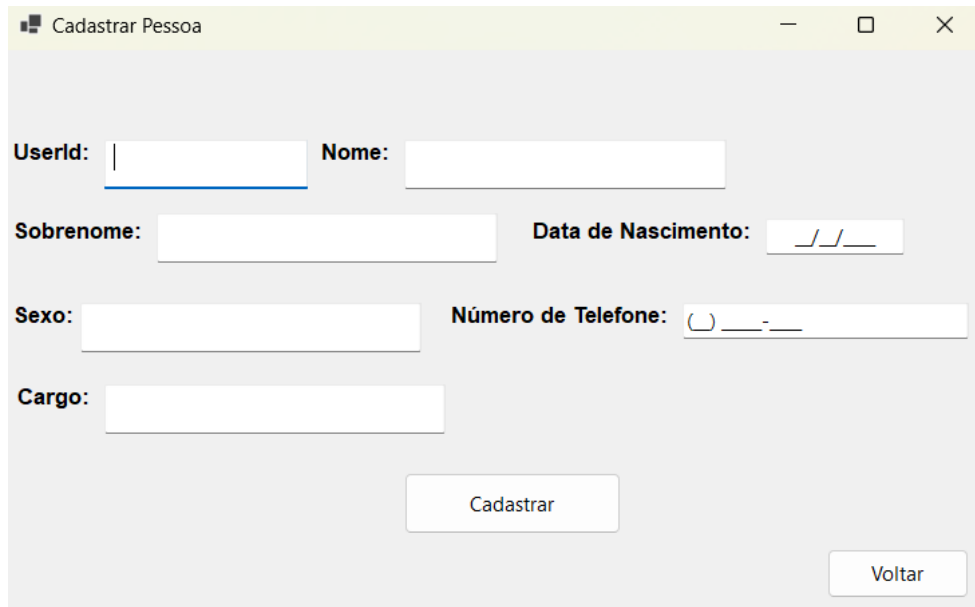
- **Cadastro de lista:** Apresenta as seguintes funcionalidades: cadastro de uma lista nova, apagar lista já incluída no programa, ordenar lista e buscar por uma lista desejada.



- **Gerenciamento de pessoas:** Apresenta as seguintes funcionalidades: Cadastro de novas pessoas, pesquisa de pessoas pelo seu código de usuário e cargo, atualização e exclusão de pessoas.

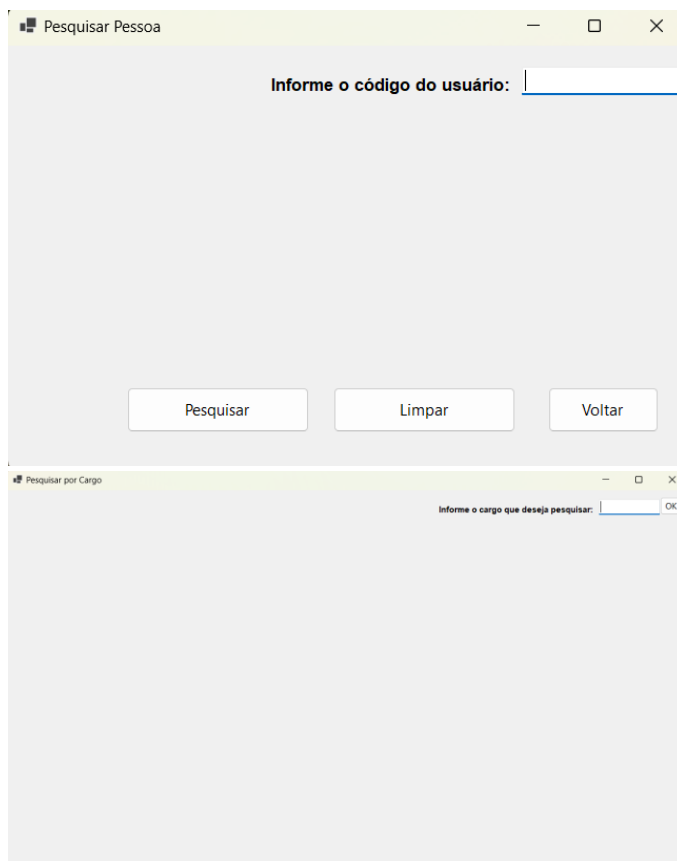


- **Cadastro de pessoa:** Possui os seguintes campos a serem inseridos: UserId (código do usuário), nome, sobrenome, data de nascimento, sexo, número de telefone e cargo ao qual a pessoa é responsável.



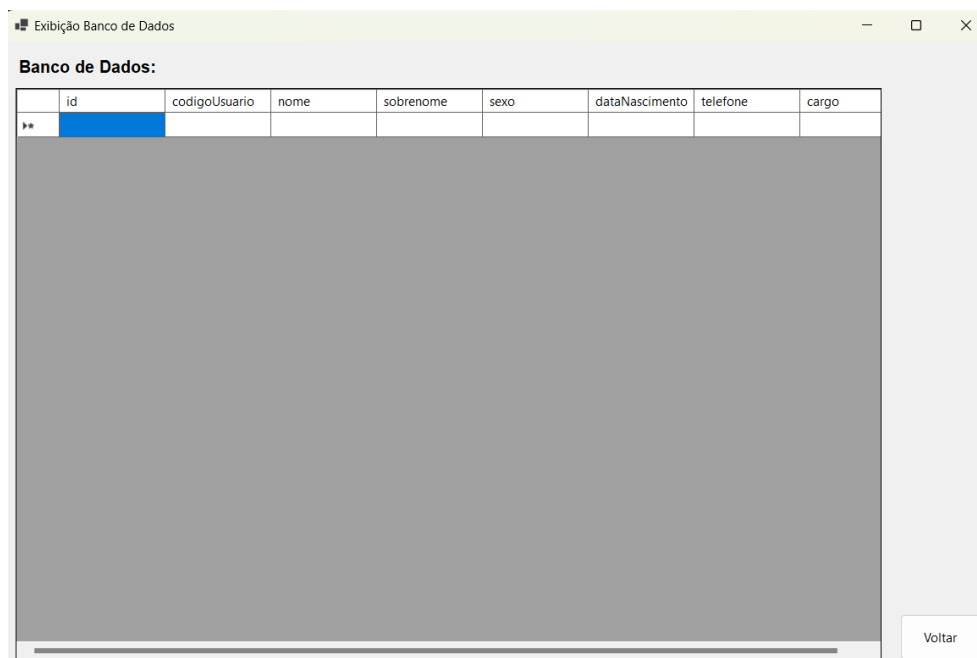
The screenshot shows a window titled "Cadastrar Pessoa" with a light gray background. It contains several input fields for user registration: "UserId:" with a text box, "Nome:" with a text box, "Sobrenome:" with a text box, "Data de Nascimento:" with a date picker showing "\_\_/\_\_/\_\_", "Sexo:" with a text box, "Número de Telefone:" with a text box containing a parenthesis and dashes, and "Cargo:" with a text box. At the bottom center is a "Cadastrar" button, and at the bottom right is a "Voltar" button.

- **Pesquisas:** Permite o usuário pesquisar pessoas informando seu código de usuário e cargo.



The screenshot shows two windows. The top window is titled "Pesquisar Pessoa" and contains a label "Informe o código do usuário:" followed by a text box. Below the text box are three buttons: "Pesquisar", "Limpar", and "Voltar". The bottom window is titled "Pesquisar por Cargo" and contains a label "Informe o cargo que deseja pesquisar:" followed by a text box and an "OK" button.

- **Visualização da lista:** Permite o usuário visualizar a lista completa de pessoas já integrada com o banco de dados e com todos dados inseridos anteriormente.



## 4.2 Manipulação de Dados

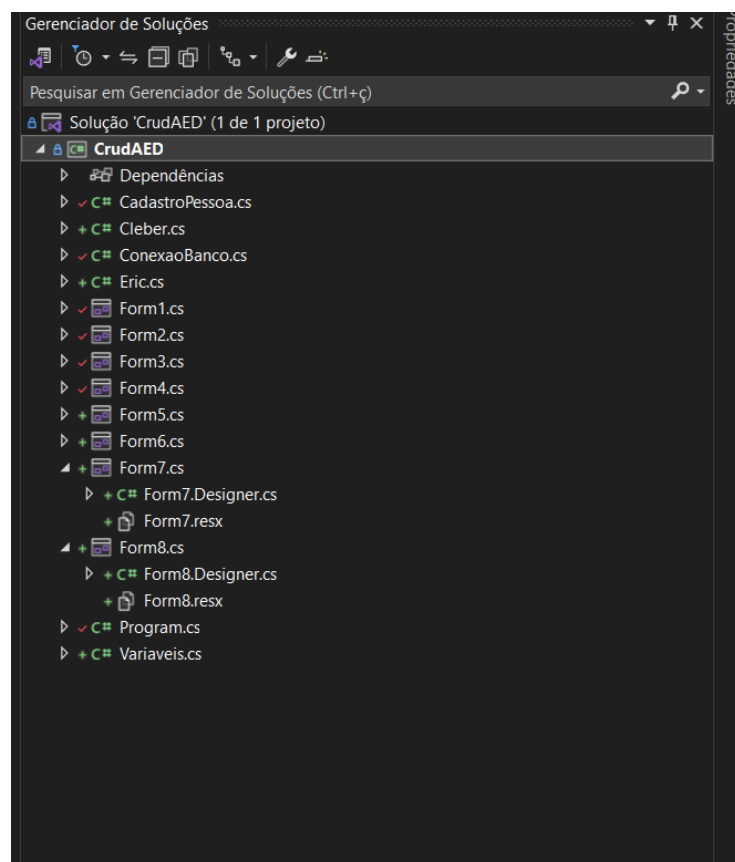
- **Carga de Arquivos:** A aplicação é capaz de carregar arquivos CSV de diferentes tamanhos, permitindo assim o gerenciamento de grandes volumes de dados.
- **CRUD de Pessoas:** A implementação das operações de criar, ler, atualizar e excluir dados de pessoas no banco de dados foi realizada de forma a garantir a eficiência e a segurança das transações.
- **Pesquisa e Ordenação:** Além do CRUD, a aplicação permite realizar pesquisas por diferentes critérios, como código do usuário e cargo, utilizando algoritmos de busca eficientes. A ordenação dos dados é feita através do algoritmo QuickSort, que foi escolhido por sua eficiência em grandes conjuntos de dados.
- **Banco de Dados:** A estrutura do banco de dados foi projetada para suportar as operações de CRUD e pesquisas de forma eficiente. A tabela pessoas inclui campos como id, codigoUsuario, nome, sobrenome, sexo, dataNascimento, telefone, e cargo, todos com tipos de dados adequados para garantir a integridade e a consistência das informações. A utilização de AUTO\_INCREMENT no campo id assegura a geração automática de identificadores únicos para cada registro. Além disso, o uso de VARCHAR permite armazenar dados de forma flexível e otimizada.

### 4.3 Estruturas de Dados

Para a manipulação eficiente dos dados, foram utilizadas as seguintes estruturas de dados:

- **Classes e Objetos:** Cada pessoa é representada como um objeto, com atributos correspondentes aos dados do CSV.
- **Listas Encadeadas:** As listas encadeadas foram utilizadas para armazenar as pessoas de forma dinâmica, permitindo a inserção e remoção eficientes. A escolha das listas encadeadas se deu pela necessidade de gerenciar listas de tamanhos variáveis, sem a sobrecarga de alocação estática de memória.
- **QuickSort:** O método de ordenação QuickSort foi implementado para ordenar as listas de pessoas de maneira eficiente. O QuickSort foi escolhido por sua complexidade de tempo  $O(n \log n)$  em casos médios e pela facilidade de implementação com listas encadeadas.
- **Banco de Dados MySQL:** A integração com o MySQL foi fundamental para armazenar os dados de forma persistente e realizar operações de CRUD de maneira robusta.

A seguir são apresentadas as imagens dos códigos que ilustram a implementação das estruturas de dados e algoritmos descritos anteriormente:



```
CrudAED
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace CrudAED
8 {
9     12 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
10     static class ConexaoBanco
11     {
12         private const string servidor = "localhost";
13         private const string bancoDados = "aed";
14         private const string usuario = "root";
15         private const string senha = "JesusIsKing01";
16
17         //declarando uma variável para fazer a conexão do banco de dados
18         static public string bancoServidor = $"server={servidor};user id={usuario};database={bancoDados};password={senha}";
19     }
20 }
```

```
CrudAED
15
16 namespace CrudAED
17 {
18     16 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
19     internal class CadastroPessoa
20     {
21         private int id;
22         private string userID;
23         private string nome;
24         private string sobrenome;
25         private string sexo;
26         private string dataNascimento;
27         private string telefone;
28         private string cargo;
29
30         2 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
31         public int Id
32         {
33             get { return id; }
34             set { id = value; }
35         }
36
37         5 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
38         public string UserID
39         {
40             get { return userID; }
41             set { userID = value; }
42         }
43
44         3 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
45         public string Nome
46         {
47             get { return nome; }
48             set { nome = value; }
49         }
50
51         2 referências | victorlcs, há 3 dias | 1 autor, 1 alteração
52         public string Sobrenome
53         {
54             get { return sobrenome; }
55             set { sobrenome = value; }
56         }
57     }
58 }
```

```
CrudAED
69
70 //Método para cadastrar pessoas ao banco de Dados
71 1 referência | victorlcs, há 3 dias | 1 autor, 1 alteração
72 public bool cadastrarPessoa()
73 {
74     try
75     {
76         MySqlConnection MySqlConexaoBanco = new MySqlConnection(ConexaoBanco.bancoServidor);
77         MySqlConexaoBanco.Open();
78
79         string insert = $"insert into pessoas (codigoUsuario,nome,sobrenome,sexo,dataNascimento,telefone,cargo) values ('{Use
80         MySqlCommand comandoSql = MySqlCommand.CreateCommand();
81         comandoSql.CommandText = insert;
82
83         comandoSql.ExecuteNonQuery();
84         return true;
85     }
86     catch (Exception ex)
87     {
88         MessageBox.Show("Erro no banco de dados - Método cadastrarPessoa:" + ex.Message);
89         return false;
90     }
91 }
92
93 1 referência | 0 alterações | 0 autores, 0 alterações
94 public bool cadastrarLista(ListaDupla x)
95 {
96     MySqlConnection MySqlConexaoBanco = null;
97     MySqlTransaction transaction = null;
98
99     try
100     {
101         MySqlConnection MySqlConexaoBanco = new MySqlConnection(ConexaoBanco.bancoServidor);
102         MySqlConexaoBanco.Open();
103         transaction = MySqlConexaoBanco.BeginTransaction();
104
105         string insert = "INSERT INTO pessoas (codigoUsuario, nome, sobrenome, sexo, dataNascimento, telefone, cargo) " +
106     }
```



```

1 • create database aed
2   default character set utf8mb4
3   default collate utf8mb4_general_ci;
4
5 • use aed;
6
7 • create table if not exists pessoas(
8     id int not null auto_increment primary key,
9     codigoUsuario varchar(10) not null,
10    nome varchar(10) not null,
11    sobrenome varchar(10),
12    sexo varchar(10),
13    dataNascimento varchar(10),
14    telefone varchar(12),
15    cargo varchar(15)
16  ) default char set utf8mb4;
17
18 • describe pessoas;
19
20 • alter table pessoas
21   modify column codigoUsuario varchar(255);

```

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

```
1 • create database aed
2   default character set utf8mb4
3   default collate utf8mb4_general_ci;
4
5 • use aed;
6
7 • create table if not exists pessoas(
8   id int not null auto_increment primary key,
9   codigoUsuario varchar(10) not null,
10  nome varchar(10) not null,
11  sobrenome varchar(10),
12  sexo varchar(10),
13  dataNascimento varchar(10),
14  telefone varchar(12),
15  cargo varchar(15)
16 ) default char set utf8mb4;
17
18 • describe pessoas;
19
20 • alter table pessoas
21   modify column codigoUsuario varchar(255);
```

## 5. Conclusão

O desenvolvimento deste projeto proporcionou uma oportunidade de aplicar os conceitos aprendidos em Algoritmos e Estruturas de Dados na prática. Através da implementação de uma aplicação para gerenciamento de dados de pessoas, pudemos explorar diferentes estruturas de dados e técnicas de programação, além de aprimorar nossas habilidades de desenvolvimento de software. O uso de listas encadeadas, algoritmos de ordenação como o QuickSort, e a integração com um banco de dados MySQL, mostrou-se crucial para atingir os objetivos de eficiência e funcionalidade do projeto.

## 6. Links Externos

Para visualizar o funcionamento da aplicação, assista ao vídeo demonstrativo disponível em: <https://youtu.be/AC39v7Sr7B4>

## REFERÊNCIAS

**Ambler, Scott W.** Análise e projeto orientados a objeto, volume II: seu guia para desenvolver sistemas robustos com tecnologia de objetos / Scott W. Ambler; tradução Oswaldo Zanelli. - Rio de Janeiro: Infobook, 1998.

**Gamma, Erich** Padrões de projeto: soluções reutilizáveis de software orientado a objetos / Erich Gama, Richard Helm, Ralph Johnson e John Vlissides; trad. Luiz A. Meirelles Salgado. - Porto Alegre: Bookman, 2000.

**Date, C J.**, 1941 - Introdução a sistemas de banco de dados / C. J. Date: tradução de Daniel Vieira. - Rio de Janeiro: Elsevier, 2003.